
US Department of Education



FAFSA on the Web Redesign Detail Design Document: End of Entry Edits

Version 1.0

FAFSA on the Web Redesign	Version: 1.0
Detail Design Documentation	Date: 3/29/2001

Revision History

Date	Version	Description	Author
4/19/2001	1.0	Detail Design – End of entry edits	David Williams

FAFSA on the Web Redesign	Version: 1.0
Detail Design Documentation	Date: 3/29/2001

Table of Contents

1	Introduction	4
2	Object Definitions	4
3	Process Flow Details	8
4	Use Case References	8

FAFSA on the Web Redesign	Version: 1.0
Detail Design Documentation	Date: 3/29/2001

Detail Design Document: End of Entry Edits

1 Introduction

These are the common specifications for the End of Entry Edits as part of the web application, **FAFSA on the Web**. These specifications apply solely to the End of Entry Edits, as determined by 2001-2002 requirements.

2 Object Definitions

Questions interface

This interface assigns a constant to each question that appears in the application process. These constants are used to reference the static data that is used to display the question to the user if an error is found during the EOE process. The int value assigned to the question may not correspond to the question number in the application.

FormConstants interface

This interface extends the Questions interface to make use of the question constants. It holds the following:

- Arrays for each edit that displays questions. Each array returns a list of question numbers that it should display for a given edit.
- A list of messages to display to the user to explain the error in their application data. There is one for each edit that displays questions.
- A set of constants to define the type of input box the question uses.
- A set of constants to define the edit numbers.
- A set of constants to define the types of options that correspond to a question in corrections.

QuestionBean object

The question bean is the template to hold the information for each question so it can be dynamically displayed back to the user. It holds the following information about the question:

- (String) lOption – the option type to be displayed to the user (for corrections only).
- (int) iType – the type of input used to accept the information from the user (textbox, radiobutton, etc.)
- (String) szDescription – the description that appears before the input box when the question is displayed. Includes the entire html.
- (String) szFieldName – the field that stores the data in the form apply bean
- (String) szHTMLTag – the html to display the input box to the user
- (String) szTitle – the question text. The text that appears on the left hand cell of the table when the question is displayed. Includes the entire html.
- (String) szValidationJavaScript – the java script required to do any checks before the form is submitted.

ValidateEOE object

This class holds all the information to check the accuracy of the data that was entered during the apply process. It holds individual functions for each edit check (as found in the change log document log.910 [and in the Appendix to this document](#)). There are functions that run each of the groups of edits as a whole (0000, 1000, 2000, 3000, 4000, and 5000 edits) that are used to run the end of entry edits when the application is submitted.

There are 3 types of edits that are used:

- confirmation – if the conditions for the edit are met a confirmation flag is checked. If it is true, the data is accepted. If it is false an exception of type confirmation is raised.

FAFSA on the Web Redesign	Version: 1.0
Detail Design Documentation	Date: 3/29/2001

- assumption – if the conditions for the edit are met some data is set (assumed to be true).
- error – if the conditions for the edit are met an exception is raised.

Any exception that is raised returns the edit number back so that information for that edit can be looked up.

ValidateEOE also has the following parameters that are used in the EFC calculation:

- (Boolean) bAutoZero – is set if the conditions are met that the EFC should be 0.
- (Boolean) bSimple – a Boolean that needs to be passed to the EFC calculation
- (int) incomeHigh – the high income value that is used in some of the edits
- (int) incomeLow – the low income value that is used in some of the edits
- (int) iParFiler – the type of tax return filed by the parents
- (int) iStudFiler – the type of tax return filed by the student

The object has a function called run that starts the EOE process. It takes in the filled out formApplyBean and runs the correct edits based on the data. If any condition is met, one of the above edit types is used to validate the data. It either is assumed or it is returned to the user to make changes to their inputs.

FormBean object

The form bean is the central repository for all the question information. It has methods to access all the information based on question or edit number. All the methods are static to increase the performance when all users call the bean. The methods are:

- displayAnswer – displays the answer input as it appears in the form apply process based on a question ID. Includes the html
- displayAnswerReadonly – displays the answer for some questions in a readonly manner based on a Question ID. Includes the html.
- displayDescription – displays the description based on a question ID. Includes html
- displayFieldName – returns the name of the field for the given question
- displayJavaScript – returns the script to populate the given question client side.
- displayQuestion – returns the question for the given question ID. Includes html.
- displayValidateJS – returns the javascript that will run when the form is submitted to validate the various fields. It may be blank for any given field.
- getConfirmation – returns the position in the confirmation array for a given edit. (See below)
- getMessage – returns the message to display to the user for a given edit.
- getOption – returns the option type for a given question. Used for corrections.
- getPage – returns the page that a given question is found on. It contains logic to look at values in the formApplyBean to return the appropriate page based on what the user has already entered.
- getQuestions – returns an array of ints that contains the questions that need to be displayed for a given edit.
- saveQuestion – makes the appropriate change in the formApplyBean based on a given question ID. It takes the value out of the request object and copies it over the value in the formApplyBean.

There is one field, aQuestions that is an array of question beans. The class statically initializes this array with a question bean for each of the questions in the application. These can then be accessed by the question ID using the methods outlined above.

EOEController object

This servlet is the controller to run the EOE process. It looks at the incoming information from the request object and steps through the logic accordingly. The flow details will be outlined below and a process flow diagram is found in section 3 of this document.

The object has the following methods:

- doPost – the entry point to the servlet. It directs the flow to the correct method.

FAFSA on the Web Redesign	Version: 1.0
Detail Design Documentation	Date: 3/29/2001

- doEOEConfirmation – this checks to see the steps in dealing with a confirmation edit. The user has the following options from the confirmation page:
 - Move back to the form pages.
 - Update the confirmation flag. In other words, verifying that the data is actually correct.
- doEOEEdit – this checks the options for dealing with an error edit. The user has the following options from the error page:
 - Move back to the form pages
 - Update the value in the formApplyBean and run through the EOE again.
- exitEOE – resets the appropriate flags for the navigation state
- redirectToErrorPage – used when an error occurs in the process.
- returnToForm – This returns the user to the appropriate page in the form if that is what they chose to do.
- RunEFC – runs the EFC if the EOE was successful and updates the formApplyBean.
- RunEOE – runs the data through the EOE validation and checks for errors.

EOE Process Flow

doPost

When the user comes into the starting point of the process, the servlet looks for the 'page' parameter. If there is no page parameter then the runEOE method is called to begin checking the data. If there is a 'page' parameter (meaning the user is coming from the EOE edit pages) then the servlet looks for an editNumber in the formApplyBean. If there are no errors, the 'page' parameter is checked to see what type of edits the user was checking (verification or error) and the correct method is called accordingly.

runEOE

This method runs the validation on the data and checks to see if it is OK. A validateEOE object is created and the run method of the object is executed. If there was an exception, the edit type (confirmation or error) is checked and the edit number is extracted. The user is directed to the correct page depending on the type of edit. The EOE.jsp page is called for an error and the EOEVERIFICATION for a confirmation. If there was no exception, the runEFC method is called to calculate the EFC value. Finally, if the EFC runs without an error, the exitEOE is called to reset the navigation flags and the user moves on to the next stage in the process.

doEOEConfirm

This method checks runs when the user comes back from the EOEVERIFICATION.jsp page. It checks the 'action' parameter to find out what the user has done. If the user chose edit the questions, they are directed to the EOE.jsp page that gives them an editable version of the questions. If the user chooses OK, they have checked the data and have confirmed that it is correct. The confirmation flag for that edit is set to 1 and the runEOE method is called to continue in the verification of the data.

doEOEEdit

This method checks the 'action' parameter to see what the user has decided to do. If the action is NEXT the user has chosen to edit the data and save it. To save the data the questions are looped through and calling the saveQuestion method of the FormBean saves the values. If the action is PREV the user has chosen to return to the form. The page belonging to the first question sequentially is the page that the user needs to be returned. To get this the array of pages is returned and the first question out of the array is found. Calling the getPage method of the formBean and passing in the page number returns the correct page. The user is redirected to the page.

returnToForm

This method takes in a page ID and directs the user to the respective page.

FAFSA on the Web Redesign	Version: 1.0
Detail Design Documentation	Date: 3/29/2001

exitEOE

This method resets the navigation flags that were used in the EOE navigation.

runEFC

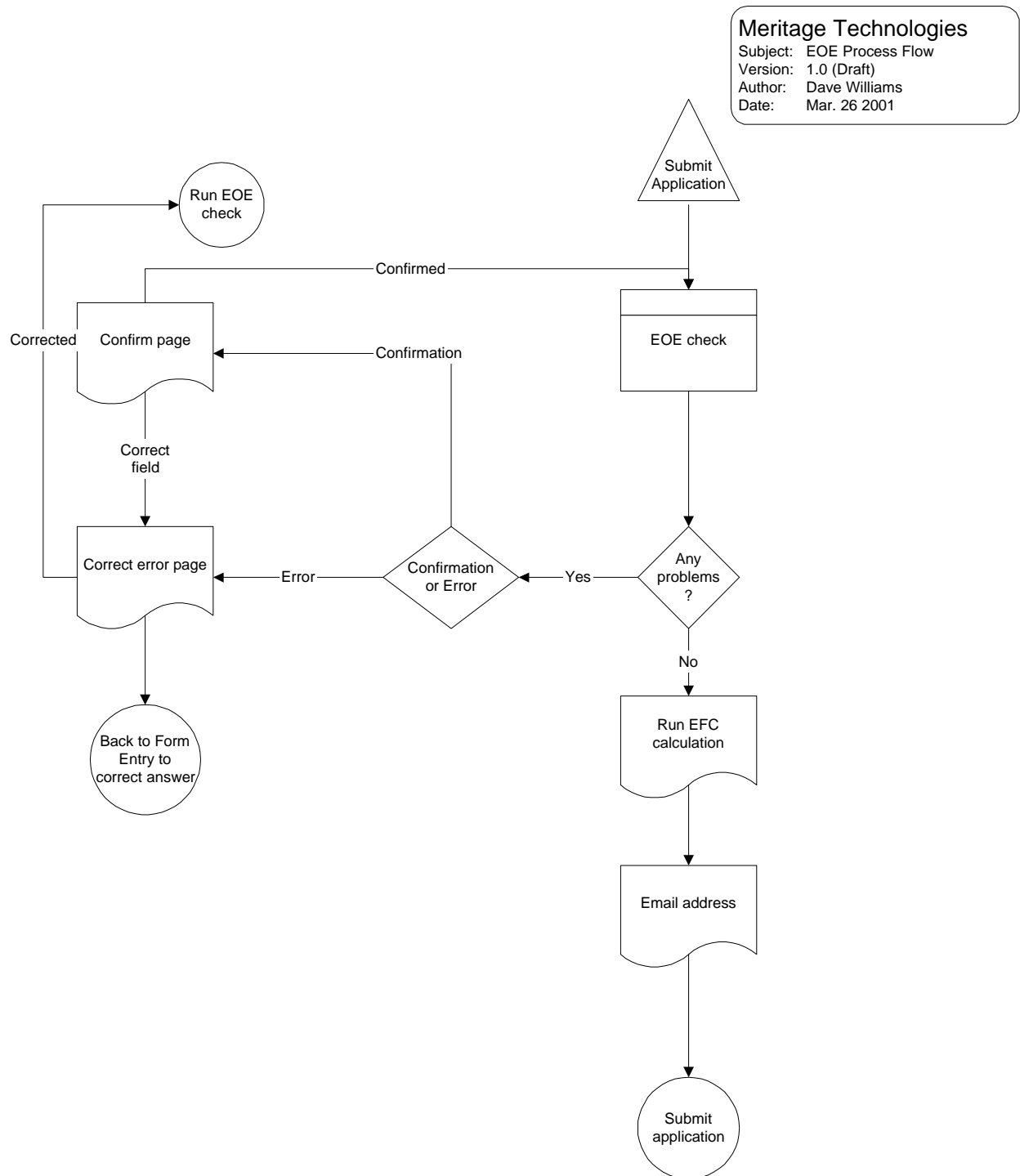
This method runs the EFC calculation and adds the value to the formApplyBean. See EFC for more details.

redirectToErrorPage

This method is called when there is an error so the user can see an error page to explain the error.

FAFSA on the Web Redesign	Version: 1.0
Detail Design Documentation	Date: 3/29/2001

3 Process Flow Details



4 Use Case References

- Use Case Specification: End of Entry Edits

5 Appendix

FAFSA on the Web Redesign	Version: 1.0
Detail Design Documentation	Date: 3/29/2001

Additional Edits

<u>Edit Number</u>	<u>Edit Condition</u>	<u>Edit Result</u>
<u>0001</u>	<u>Student is dependent and Parent Marital status is blank</u>	<u>Raise an error and display the question for parent marital status</u>
<u>0002</u>	<u>Student is dependent and Parent number in college is blank</u>	<u>Raise error and display the question for parent number in college</u>
<u>0003</u>	<u>Student is dependent and Parent tax form type is blank</u>	<u>Raise error and display the question for parent tax form type.</u>